An Efficient Constraint Force Computation in Multibody Systems*

Sung-Soo KIM** and Jin Young YOU***

An efficient method for constraint force analysis of specified joints has been developed in a serially connected open chain multibody system. In the method, the system equations of motion are derived using the recursive formulation, but constraint forces are computed non-recursively for specified joints. The efficiency of the proposed formulation has been compared with the recursive Newton-Euler formulation by the operational counting and the CPU time measure. To validate solution accuracy of the proposed method, the 7 d.o.f RRC robot arm simulation has been carried out. Essentially the same simulation results are obtained from the proposed method and from the commercial dynamics analysis program DADS.

Key Words : Recursive Multibody Dynamics, Constraint Force, Joint Relative Coordinates

1. Introduction

The determination of constraint forces due to joints in multibody systems is important for durability analysis in the part design. In computational fatigue analysis, the history of dynamic loads such as constraint reaction forces and inertia forces must be computed from multibody dynamic analysis.

Several formulations based on maximal (Cartesian or absolute) coordinates and Lagrange multipliers lead to DAE (Differential-Algebraic Equations) form of equations of motion^{(1),(2)}. These formulations express constraint forces in terms of Lagrange multipliers, which can be obtained directly from DAEs as a part of the solution. However, the formulations related to maximal coordinates are computationally inefficient, since they have to deal with large size of DAEs. On the other hand, the formulations based on minimal (joint relative) coordinates often lead to ODE (Ordinary Differential Equations) form of equations of motion. Although they are computationally

efficient to solve equations of motion, they exclude the constraint reaction forces, since the coordinates used in their formulations are already in the tangent plane of the constraint surface⁽³⁾.

To determine constraint forces in the minimal (joint relative) coordinate formulations, several approaches have been studied. There are two distinctive approaches; one is the projection method⁽⁴⁾ using constraint Jacobian matrix, and the other is the recursive Newton-Euler method⁽⁵⁾, which leads to D'Alembert dynamic equilibrium equations in each body. In the projection method, the recent development for constraint force determination is presented in Ref. (6). 'Open constraint coordinates' are introduced. Using constraint Jacobian matrix associated with the open constraint coordinates, generalized forces are projected onto the directions related to the open constraint coordinates. In the recursive Newton-Euler method, Cartesian accelerations at the center of mass in each body have been recovered, after solving equations of motion in the minimal coordinate space. With known body center's acceleration, constraint reaction force is computed recursively from tree end body to the base body in open topology system, using D'Alembert dynamic equilibrium equations.

In this paper, another efficient approach for constraint force determination is proposed, based on a variation of the recursive Newton-Euler equations.

* Received 15th October, 2002 (No. 02-5169)

** Dept. of Mechatronics Engineering, Chungnam National University, 220 Kung-dong, Yusung-ku, Daejeon, 305-764, Korea. E-mail: sookim@cnu.ac.kr

*** Seoul National University, San 56-1, Shinlim-dong, Kwanak-ku, Seoul, 151-742, Korea. E-mail: jean@icat.snu.ac.kr

Although the recursive formulation^{(7),(8)} is employed to generate equations of motion, constraint reaction forces are non-recursively computed only for specified joints, making use of already computed terms in the recursive formulation.

2. Multibody Equations of Motion based on the Recursive Formulation

2.1 Recursive kinematics

To define kinematics of a multibody system, a pair of rigid bodies connected by a joint is shown in Fig. 1. The primed reference frames represent the body fixed reference frames and the double primed reference frames express the joint reference frames, respectively. The orientation of the body j reference frame with respect to the inertial reference frame can be represented by the following sequence of direction cosine matrices.

$$\mathbf{A}_j = \mathbf{A}_i \mathbf{C}_{ij} \mathbf{A}_{ij}'' \quad (1)$$

where $\mathbf{A}_i$ and $\mathbf{A}_j$ are the orientation matrices of the body i and j , respectively, $\mathbf{C}_{ij}$ is the orthogonal transformation matrix from the $x_{ij}''-y_{ij}''-z_{ij}''$ frame to the $x_i'-y_i'-z_i'$ frame. $\mathbf{A}_{ij}''$ is the orthogonal transformation matrix from the $x_j'-y_j'-z_j'$ frame to the $x_{ij}''-y_{ij}''-z_{ij}''$ frame. The position vector $\mathbf{r}_j$ of the body j can be written as;

$$\mathbf{r}_j = \mathbf{r}_i + \mathbf{s}_{ij} + \mathbf{d}_{ij} = \mathbf{r}_i + \mathbf{A}_i \mathbf{s}_{ij}' + \mathbf{A}_i \mathbf{C}_{ij} \mathbf{d}_{ij}''(\mathbf{q}_j) \quad (2)$$

where $\mathbf{r}_i$ is the position vector of the body i , and the position vector $\mathbf{s}_{ij}$ is the vector from the origin of the body i reference frame $\mathbf{O}_i'$ to that of the joint reference frame $\mathbf{O}_{ij}''$, and the position vector $\mathbf{d}_{ij}$ is the vector from the origin of the joint reference frame $\mathbf{O}_{ij}''$ to that of the body j reference frame $\mathbf{O}_j'$. It is noted that $\mathbf{d}_{ij}$ is a function of relative joint coordinate $\mathbf{q}_j$ ⁽⁸⁾.

Fig. 1 A pair of connected bodies

Angular velocity of body j can be expressed in terms of the angular velocity of body i and the relative angular velocity of body j with respect to body i as;

$$\boldsymbol{\omega}_j = \boldsymbol{\omega}_i + \boldsymbol{\omega}_{ij} = \boldsymbol{\omega}_i + \mathbf{H}_j \dot{\mathbf{q}}_j \quad (3)$$

where $\boldsymbol{\omega}_{ij}$ is the relative angular velocity of the body j with respect to the body i , and is also function of joint relative coordinate $\mathbf{q}_j$ and joint relative velocity $\dot{\mathbf{q}}_j$. $\mathbf{H}_j$ is the unit vector of the joint rotational axis.

Translational velocity relationship between body j and body i can be obtained by differentiating Eq. (2). Substituting the expression obtained from Eq. (3) into the resulting equations yields

$$\dot{\mathbf{r}}_j + \tilde{\mathbf{r}}_j \boldsymbol{\omega}_j = \dot{\mathbf{r}}_i + \tilde{\mathbf{r}}_i \boldsymbol{\omega}_i + \left(\frac{\partial \mathbf{d}_{ij}}{\partial \mathbf{q}_j} + \tilde{\mathbf{r}}_j \mathbf{H}_j \right) \dot{\mathbf{q}}_j \quad (4)$$

where the tilde operator is defined as

$$\tilde{\mathbf{r}} = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \quad (5)$$

In order to have more compact form of the equations, the composite state vector notation can be employed⁽⁸⁾. Using Eqs. (3) and (4), the compact form of velocity relationship between two bodies is obtained in the state vector form as

$$\hat{\mathbf{Y}}_j = \hat{\mathbf{Y}}_i + \mathbf{B}_j \dot{\mathbf{q}}_j \quad (6)$$

where, $\hat{\mathbf{Y}}_j = \begin{bmatrix} \dot{\mathbf{r}}_j + \tilde{\mathbf{r}}_j \boldsymbol{\omega}_j \\ \boldsymbol{\omega}_j \end{bmatrix}$ and $\hat{\mathbf{Y}}_i = \begin{bmatrix} \dot{\mathbf{r}}_i + \tilde{\mathbf{r}}_i \boldsymbol{\omega}_i \\ \boldsymbol{\omega}_i \end{bmatrix}$ are the velocity state vectors of bodies i and j , respectively, and $\mathbf{B}_j = \begin{bmatrix} \frac{\partial \mathbf{d}_{ij}}{\partial \mathbf{q}_j} + \tilde{\mathbf{r}}_j \mathbf{H}_j \\ \mathbf{H}_j \end{bmatrix}$ is the velocity transformation

matrix from the joint space to the state vector space.

The Cartesian velocity for body j can be recovered, once $\hat{\mathbf{Y}}_j$ is obtained; i.e.,

$$\mathbf{Y}_j = \mathbf{T}_j^{-1} \hat{\mathbf{Y}}_j \quad (7)$$

where $\mathbf{T}_j^{-1}$ and $\mathbf{Y}_j$ are defined as $\mathbf{T}_j^{-1} = \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{r}}_j \\ 0 & \mathbf{I} \end{bmatrix}$ and

$\mathbf{Y}_j = \begin{bmatrix} \dot{\mathbf{r}}_j \\ \boldsymbol{\omega}_j \end{bmatrix}$, respectively.

The acceleration state vector of body j is also obtained by differentiating Eq. (6); i.e.,

$$\begin{aligned} \ddot{\mathbf{Y}}_j &= \ddot{\mathbf{Y}}_i + \mathbf{B}_j \ddot{\mathbf{q}}_j + \dot{\mathbf{B}}_j \dot{\mathbf{q}}_j \\ &\equiv \ddot{\mathbf{Y}}_i + \mathbf{B}_j \ddot{\mathbf{q}}_j + \mathbf{D}_j \end{aligned} \quad (8)$$

The relationship between the acceleration state vector and the Cartesian acceleration vector can be obtained by differentiating Eq. (7) as

$$\begin{aligned} \ddot{\mathbf{Y}}_j &= \mathbf{T}_j^{-1} \ddot{\mathbf{Y}}_j + \dot{\mathbf{T}}_j^{-1} \dot{\mathbf{Y}}_j \\ &= \mathbf{T}_j^{-1} \ddot{\mathbf{Y}}_j - \mathbf{R}_j \end{aligned} \quad (9)$$

To explain position, velocity, and acceleration analysis of a rigid multibody system, using recursive kinematics, a serially connected open chain system is shown in Fig. 2. Once the base body (body 0) position and orientation, relative joint coordinates of the open chain system are given, then the Cartesian position

Fig. 2 A serially connected open loop subsystem

and orientation of the each body can be recursively obtained using Eqs. (1) and (2) from the base body 0 to the tree end body n . In the same fashion, the velocity and the acceleration states of each body in the open chain system are also recursively obtained using Eqs. (6) and (8), respectively.

2.2 Equations of motion for an open chain system

Figure 2 represents a schematic diagram of an open chain system that comprises of $n+1$ rigid bodies and n joints. The variational form of equations of motion for this open chain system is obtained by adding virtual work of each body as follows⁽⁸⁾;

$$\sum_{i=0}^n \delta \bar{Z}_i^T (\bar{M}_i \dot{\bar{Y}}_i - \bar{Q}_i) = 0 \quad (10)$$

where $\delta \bar{Z}_i = \begin{bmatrix} \delta \mathbf{r}_i + \tilde{\mathbf{r}}_i \delta \boldsymbol{\pi}_i \\ \delta \boldsymbol{\pi}_i \end{bmatrix}$ is the kinematically admissible generalized virtual displacements associated with the state vector defined in Eq. (6). $\delta \mathbf{r}_i$ and $\delta \boldsymbol{\pi}_i$ are the virtual displacement and virtual rotation, respectively. $\bar{M}_i$ is the generalized mass matrix in state vector space and $\bar{Q}_i$ is the generalized force vector in the state vector space. They are defined as;

$$\bar{M}_i = \begin{bmatrix} m_i \mathbf{I} & -m_i \tilde{\mathbf{r}}_i^c \\ m_i \tilde{\mathbf{r}}_i^c & \mathbf{J}_i - m_i \tilde{\mathbf{r}}_i^c \mathbf{r}_i^c \end{bmatrix} \quad (11)$$

$$\bar{Q}_i = \begin{bmatrix} \mathbf{f}_i^c + m_i \dot{\tilde{\mathbf{r}}}_i^c \boldsymbol{\omega}_i \\ \mathbf{n}_i^c + \tilde{\mathbf{r}}_i^c \mathbf{f}_i^c + m_i \tilde{\mathbf{r}}_i^c \dot{\tilde{\mathbf{r}}}_i^c \boldsymbol{\omega}_i - \tilde{\boldsymbol{\omega}}_i \mathbf{J}_i \boldsymbol{\omega}_i \end{bmatrix} \quad (12)$$

where m_i is the mass of the body i , $\mathbf{J}_i^c$ is the moment of inertia matrix of body i with respect to the centroidal body reference frame, and $\mathbf{r}_i^c$ is the position vector to the center of mass of the body i . In Eq. (12), $\mathbf{f}_i^c$ is the resultant applied force that is acting at the center of mass of the body i , $\mathbf{n}_i^c$ is the resultant applied moment acting on the body i . In order to transform the variational form of equations of motion defined in state vector space into the one in the joint space using recursive formulation, the variational relationship of the state vectors associated with the bodies i and $i-1$ is defined as;

$$\delta \bar{Z}_i = \delta \bar{Z}_{i-1} + \mathbf{B}_i \delta \mathbf{q}_i \quad (13)$$

where $\delta \mathbf{q}_i$ is the virtual displacement of the joint i .

The acceleration state vector of the each body is obtained in terms of the acceleration state vector of the base body 0 and the joint accelerations in the open chain system, if Eq. (8) is applied recursively; i.e.,

$$\begin{aligned} \dot{\bar{Y}}_i &= \dot{\bar{Y}}_{i-1} + \mathbf{B}_i \ddot{\mathbf{q}}_i + \mathbf{D}_i \\ &\equiv \dot{\bar{Y}}_{i-2} + \sum_{k=i-1}^i (\mathbf{B}_k \ddot{\mathbf{q}}_k + \mathbf{D}_k) \\ &\vdots \\ &\equiv \dot{\bar{Y}}_0 + \sum_{k=1}^i (\mathbf{B}_k \ddot{\mathbf{q}}_k + \mathbf{D}_k) \end{aligned} \quad (14)$$

If Eqs. (13) and (14) are substituting into Eq. (10) from the tree end body n to the base body 0, recursively, the equations of motion of the open chain system in joint space are now obtained as;

$$\mathbf{M} \ddot{\mathbf{q}} = \mathbf{P} \quad (15)$$

where,

$$\mathbf{M} = \begin{bmatrix} \mathbf{B}_n^T \mathbf{K}_n \mathbf{B}_n & \mathbf{B}_n^T \mathbf{K}_n \mathbf{B}_{n-1} & \cdots \\ & \mathbf{B}_{n-1}^T \mathbf{K}_{n-1} \mathbf{B}_{n-1} & \cdots \\ & & \ddots \\ & & & \text{symm} \\ & \mathbf{B}_1^T \mathbf{K}_1 \mathbf{B}_1 & \mathbf{B}_1^T \mathbf{K}_1 \mathbf{B}_0 \\ & \mathbf{B}_1^T \mathbf{K}_1 \mathbf{B}_1 & \mathbf{B}_1^T \mathbf{K}_1 \mathbf{B}_0 \\ & & \mathbf{K}_0 \end{bmatrix},$$

$$\ddot{\mathbf{q}} = [\ddot{\mathbf{q}}_n, \ddot{\mathbf{q}}_{n-1}, \dots, \ddot{\mathbf{q}}_2, \ddot{\mathbf{q}}_1, \dot{\bar{Y}}_0^T]^T$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{B}_n^T (\mathbf{L}_n - \mathbf{K}_n \sum_{i=1}^n \mathbf{D}_i) \\ \mathbf{B}_{n-1}^T (\mathbf{L}_{n-1} - \mathbf{K}_{n-1} \sum_{i=1}^{n-1} \mathbf{D}_i) \\ \vdots \\ \mathbf{B}_2^T (\mathbf{L}_2 - \mathbf{K}_2 \sum_{i=1}^2 \mathbf{D}_i) \\ \mathbf{B}_1^T (\mathbf{L}_1 - \mathbf{K}_1 \mathbf{D}_1) \\ \mathbf{L}_0 \end{bmatrix}$$

During this process, composite mass matrices $\mathbf{K}_i$ and composite force vectors $\mathbf{L}_i$ shown in the system mass matrix $\mathbf{M}$ and force vector $\mathbf{P}$ can be efficiently computed⁽⁸⁾ from the backward recurrence formulation as;

$$\begin{aligned} \mathbf{K}_n &= \hat{\mathbf{M}}_n \\ \mathbf{K}_i &= \mathbf{K}_{i+1} + \hat{\mathbf{M}}_i, \quad i = n-1, n-2, \dots, 1, 0 \end{aligned} \quad (16)$$

$$\begin{aligned} \mathbf{L}_n &= \hat{\mathbf{Q}}_n \\ \mathbf{L}_i &= \hat{\mathbf{Q}}_i + (\mathbf{L}_{i+1} - \mathbf{K}_{i+1} \mathbf{D}_{i+1}) \\ &\quad i = n-1, n-2, \dots, 1, 0 \end{aligned} \quad (17)$$

3. Constraint Force Formulation for Multibody Systems

In order to explain the constraint force formulation, a free body diagram of a body in the serially connected multibody system is first shown in Fig. 3.

In Fig. 3, $\mathbf{f}'_{i,i-1}$, $\mathbf{n}'_{i,i-1}$ are the constraint force and torque vectors due to the joint that connects body i and body $i-1$. Here, the primed vector is defined as

Fig. 3 A free body diagram of a body

the vector represented in the body reference frame $x'-y'-z'$. Similarly, f'_{i+1} and n'_{i+1} are the constraint force and torque vectors acting at the joint that connects body i and body $i+1$, respectively. They are represented in the joint reference frame $x''-y''-z''$ on the body i . $f'_{i+1,i}$ and $n'_{i+1,i}$ are the constraint force and torque vectors represented in body $i+1$ reference frame, due to the joint connects between body i and body $i+1$. The constraint force and torque vectors at the joints described above can be represented in the global reference frame by the coordinate transformation as

$$f_{i,i+1} = A_i C_{i,i+1} f'_{i+1} \quad (18)$$

$$n_{i,i+1} = A_i C_{i,i+1} n'_{i+1} \quad (19)$$

$$f_{i+1,i} = A_{i+1} f'_{i+1,i} \quad (20)$$

$$n_{i+1,i} = A_{i+1} n'_{i+1,i}$$

$$f_{i,i-1} = A_i f'_{i,i-1}$$

$$n_{i,i-1} = A_i n'_{i,i-1}$$

According to the Newton's third law of motion, the constraint forces and torques acting on body i and body $i+1$, due to the joint that connects body i and body $i+1$ have the action and reaction relationship; i.e.,

$$f_{i,i+1} = -f_{i+1,i} \quad (21)$$

$$n_{i,i+1} = -n_{i+1,i}$$

To include the constraint forces and torques to the variational equations of motion in Eq.(10), the virtual work due to the constraint force can be obtained as

$$\begin{aligned} \delta W &= -(\delta r_i - \tilde{s}_i \delta \pi_i)^T f_{i,i+1} - \delta \pi_i^T n_{i,i+1} \\ &\quad - \delta r_{i+1}^T f_{i+1,i} - \delta \pi_{i+1}^T n_{i+1,i} \\ &= -\delta Z_i^T \begin{bmatrix} I & 0 \\ \tilde{s}_i & I \end{bmatrix} \begin{bmatrix} f_{i+1,i} \\ n_{i+1,i} \end{bmatrix} - \delta Z_{i+1}^T \begin{bmatrix} f_{i,i+1} \\ n_{i,i+1} \end{bmatrix} \end{aligned} \quad (22)$$

Above expression can be transformed into the one in the state vector space, using the relationship between state vectors and the Cartesian vectors $\delta Z_j = T_j^{-1} \delta \tilde{Z}_j$. Applying this relationship to Eq.(22) yields

$$\delta W = -\delta \tilde{Z}_i^T \tilde{R}_{i,i+1} - \delta \tilde{Z}_{i+1}^T \tilde{R}_{i+1,i} \quad (23)$$

$$\text{where } \tilde{R}_{i,i+1} = \begin{bmatrix} f_{i,i+1} \\ (\tilde{r}_i + \tilde{s}_i) f_{i,i+1} + n_{i,i+1} \end{bmatrix} \text{ and } \tilde{R}_{i+1,i} = \begin{bmatrix} f_{i+1,i} \\ \tilde{r}_{i+1} f_{i+1,i} + n_{i+1,i} \end{bmatrix}$$

Thus, the action and reaction relationship of the constraint force vectors represented in Eq.(21) can be also expressed more compact form in state vector space as

$$\tilde{R}_{i,i+1} = -\tilde{R}_{i+1,i} \quad (24)$$

To include constraint reaction forces into the equations of motion, the virtual work due to all the constraint forces in the open chain system can be added to Eq.(10).

$$\begin{aligned} \delta \tilde{Z}_0^T (\tilde{M}_0 \dot{\tilde{Y}}_0 - \tilde{Q}_0 - \tilde{R}_{0,1}) \\ + \sum_{i=1}^{n-1} \delta \tilde{Z}_i^T (\tilde{M}_i \dot{\tilde{Y}}_i - \tilde{Q}_i - \tilde{R}_{i,i-1} - \tilde{R}_{i,i+1}) \\ + \delta \tilde{Z}_n^T (\tilde{M}_n \dot{\tilde{Y}}_n - \tilde{Q}_n - \tilde{R}_{n,n-1}) = 0 \end{aligned} \quad (25)$$

Since the variational form of equations of motion in Eq.(25) is now including the generalized constraint forces, the virtual displacements of the state vector are arbitrary. By the orthogonality theorem⁽³⁾, all the coefficients of the virtual displacement vectors in Eq.(25) must be zero. Therefore, $n+1$ Newton-Euler equations are obtained in the state vector form.

In general, the constraint forces in the serially connected open chain system are computed from tree end body n to the base body 0 , using the Newton-Euler equations and the action-reaction relationship, in backward recurrence sequence⁽⁵⁾.

If we are only interested in dynamic load history of a certain body, we only need constraint forces associated with that body for the purpose of design. However, the recursive computation of the constraint forces based on the Newton-Euler equations must be applied to all joints in the system, due to the nature of recursive computation. In this paper, an efficient method is proposed to compute constraint forces for only specified joints without using recursive computation.

To obtain the constraint forces in state vector form, the equations of motion of the n th body are expressed using Eq.(25) as

$$\tilde{M}_n \dot{\tilde{Y}}_n - \tilde{Q}_n - \tilde{R}_{n,n-1} = 0 \quad (26)$$

Substituting Eqs.(16) and (17) into Eq.(26) and rearranging for the constraint force vector yield

$$\tilde{R}_{n,n-1} = -L_n + K_n \dot{\tilde{Y}}_n \quad (27)$$

In Eq.(27), the constraint force $\tilde{R}_{n,n-1}$ of n th joint can be computed, once the state acceleration $\dot{\tilde{Y}}_n$ of the body n is given.

The constraint force at the $(n-1)$ th joint is obtained in the similar fashion, using the Newton-Euler equations of the body $n-1$, which can be directly obtained from Eq.(25) and the action-reaction relationship in Eq.(24).

$$\hat{\mathbf{R}}_{n-1,n-2} = -\hat{\mathbf{Q}}_{n-1} + \hat{\mathbf{M}}_{n-1} \dot{\hat{\mathbf{Y}}}_{n-1} + \hat{\mathbf{R}}_{n,n-1} \quad (28)$$

The constraint force vector $\hat{\mathbf{R}}_{n-1,n-2}$ at the $(n-1)$ th joint must be computed only after obtaining the constraint force $\hat{\mathbf{R}}_{n,n-1}$ from Eq. (27). Thus, in order to compute the constraint forces at the other joints in the open chain system, the Newton-Euler equations of each body must be recursively solved from tree end body to the base body.

In order to avoid such backward recursive computation and to be free from the dependency of the outboard body joint constraint force, substituting Eqs. (27), (8), (16), and (17) into Eq. (28) and rearranging yield

$$\hat{\mathbf{R}}_{n-1,n-2} = -\mathbf{L}_{n-1} + \mathbf{K}_{n-1} \dot{\hat{\mathbf{Y}}}_{n-1} + \mathbf{K}_n \mathbf{B}_n \ddot{\mathbf{q}}_n \quad (29)$$

Thus, the constraint force, acting on body $n-1$, at the $(n-1)$ th joint is dependent upon the state acceleration of the body $n-1$ and the joint acceleration of the n th joint, and the composite force vector of the body $n-1$. By the mathematical induction, the constraint force vector, acting on body i , at the i th joint that is located between body $i-1$ and body i , is obtained as;

$$\hat{\mathbf{R}}_{i,i-1} = -\mathbf{L}_i + \mathbf{K}_i \dot{\hat{\mathbf{Y}}}_i + \sum_{j=i+1}^n \mathbf{K}_j \mathbf{B}_j \ddot{\mathbf{q}}_j \quad (30)$$

In Eq. (30), the terms $\mathbf{L}_i$, $\mathbf{K}_i$, $\mathbf{K}_j \mathbf{B}_j$ are readily available. They have been computed, when the system equations of motion in Eq. (15) are formed. The state and joint accelerations are also available after solving Eq. (15). Therefore, without having constraint forces of the joints from n to $i+1$ based on recursive computation of the Newton-Euler equations, the constraint force vector at the specified joint i can be directly computed just after solving system equations of motion. To obtain the constraint forces in the Cartesian space, the constraint force vector in the state vector space can be transformed into that in Cartesian space; i.e.,

$$\begin{bmatrix} \mathbf{f}_{i,i-1} \\ \mathbf{n}_{i,i-1} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ -\tilde{\mathbf{r}}_i & \mathbf{I} \end{bmatrix} \hat{\mathbf{R}}_{i,i-1} \quad (31)$$

Within one integration time step⁽⁹⁾, the step-by-step algorithm for constraint force analysis is given as follows;

Constraint force computational algorithm

Step 1

For given joint coordinates, which are from initial conditions or from the numerical integration scheme at the current time step, position and orientation of each body are recursively computed, using Eqs. (1) and (2) from base body 0 to the tree end body n .

Step 2

For given joint velocities, which are from initial conditions or from the numerical integration scheme at the current time step, the state velocity and the Cartesian velocity of each body can be recursively computed using Eqs. (6) and (7) from the base body

Table 1 Efficiency comparison between the proposed method and the Newton-Euler method

Operation counter	Proposed method	Newton-Euler recursive method
No. of multiplication	$6(n-i)+36$	$36(n-i)+36$
No of addition	$6(n-i)+36$	$42(n-i)+36$

0 to the tree end body n .

Step 3

Generalized mass matrix and force vector of the each body are computed by Eqs. (11) and (12).

Step 4

The composite mass and force vector of the each body are recursively computed using Eqs. (16) and (17) from tree end body n to the base body 0, and then the system equations of motion in Eq. (15) is formed.

Step 5

Joint accelerations are obtained by solving system equations of motion, using numerical method such as Gaussian elimination.

Step 6

The state acceleration and the Cartesian acceleration are computed using Eqs. (8) and (9).

Step 7

Compute constraint force for the specified joints using Eqs. (30) and (31).

To investigate efficiency of the proposed algorithm, computational amount in the proposed method is compared with that in the backward recursive Newton-Euler equations⁽⁹⁾, by counting arithmetic operators in the algorithms. Comparison with other methods based on the maximal coordinates and Lagrange multipliers is excluded, since the methods based on the maximal coordinates lead to large set of DAEs and the philosophy behind these methods is totally different. In the comparison, it is assumed that the open chain system shown in Fig. 2 consists of n revolute joints and $n+1$ rigid bodies.

In Table 1, numbers of addition and multiplication operators are compared to compute the constraint force vector for the i th joint. As we can see in the Table 1, the proposed method is efficient for the all the joints except n th joint. The computational amount to compute constraint force for n th joint is the same.

4. Constraint Force Computation for 7 Axis RRC Robot Arm

To prove the efficiency of the proposed method numerically, a program for dynamic analysis and reaction force computation for the RRC robot arm⁽¹⁰⁾ has been implemented using Fortran 77 language in SGI workstation with R4000 chip. Adams-Bashforth

3rd order integration scheme has been employed⁽⁹⁾.

Figure 4 shows the initial configuration of the RRC robot arm that consists of 8 links including the base and 7 joints. Joints 1, 3, 5, 7 are roll joints, whereas the joints 2, 4, 6 are pitch joints. The mass center of this particular RRC arm is located at the joint. The i th body and i th joint reference frames are identical in Fig. 4. Thus, only body frames are shown in Fig. 4. The inertia properties of the RRC arm are shown in Table 2.

Simulations are carried out with fixing relative d.o.f. of four roll joints and with the driver at the joint 2 (pitch joint). Initial joint angles and joint angular velocities of all other joints are zeros. The driving function is used as

Table 2 Mass and moment of inertia of RRC robot arm system

	Mass (kg)	Moment of inertia ($\text{kg} \cdot \text{m}^2$)					
		I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{yz}	I_{zx}
Link 1	104.1012	0.780491	1.13	1.13	0.0	0.0	0.0
Link 2	55.3392	0.345328	0.791	0.791	0.0	0.0	0.0
Link 3	28.8036	0.200349	0.550084	0.550084	0.0	0.0	0.0
Link 4	20.412	0.176167	0.201399	0.201399	0.0	0.0	0.0
Link 5	11.1132	0.042262	0.135261	0.135261	0.0	0.0	0.0
Link 6	5.67	0.11187	0.04181	0.04181	0.0	0.0	0.0
Link 7	3.6288	0.0226	0.0226	0.226	0.0	0.0	0.0

Fig. 4 RRC robot arm system

$$q_2(t) = -\frac{\pi}{3} \sin\left(\frac{\pi}{5}t\right) \quad (32)$$

Figure 5 shows the constraint reaction force of the joint 2 in the global x -direction. The solutions from the proposed method, the Newton-Euler recursive method, and the commercial program DADS⁽¹¹⁾ are essentially the same.

In order to see the applicability of the proposed algorithm to the closed chain system, the grabbing simulation is carried out. The RRC arm grabs the ground with compliance, a stiff spring with stiffness 3.0668×10^9 N/m is attached between the ground and the gripper. Again 4 roll joints and the 2nd joint (pitch joint) are fixed not to rotate. The driver is now applied to the 4th joint (another pitch joint) with driving function as

$$q_4(t) = \frac{\pi}{30} \sin(\pi t - \pi) \quad (33)$$

The initial conditions of all other joints are zeros. Figure 6 shows the constraint reaction force of the

Fig. 5 Joint reaction force of joint 2 in x -direction

Fig. 6 Joint reaction force of joint 4 in x -direction

Table 3 CPU time comparison between proposed method and the Newton-Euler method

	Proposed method(sec)	Newton-Euler (sec)	Ratio(Newton /Proposed)
Joint 1	0.364863e-04	0.768031e-04	2.105
Joint 2	0.352150e-04	0.732973e-04	2.081
Joint 3	0.339175e-04	0.676080e-04	1.993
Joint 4	0.345843e-04	0.624957e-04	1.807
Joint 5	0.342007e-04	0.539778e-04	1.578
Joint 6	0.336119e-04	0.417999e-04	1.244
Joint 7	0.496852e-04	0.563670e-04	1.134

joint 4 in the global x -direction. Again, essentially the same results are obtained from three different programs. The simulation results validate the solution accuracy of the proposed method.

CPU time is also measured in Table 3 to compare the actual efficiency over the conventional Newton-Euler method. Maximum average speed up ratio is about 2.1, which is slower than the theoretical speed up 3.5 obtained from Table 1. Authors speculate the difference between actual and theoretical efficiencies could be due to the overhead of the calling subroutines and number of arguments in the program. For joint 7, theoretical speed up ratio must be 1.0. However, in actual average CPU time measure, 1.134 speed up ratio is achieved. This difference is due to how average CPU time measured and time sharing condition of CPU chip.

5. Conclusions

An efficient computational algorithm to compute constraint forces for a specified joint is proposed for an open chain multibody system. In the analysis, the recursive formulation is used to form the equations of motion. Already computed terms in forming equations of motion are reused to compute constraint forces for specified joints non-recursively. Numerical efficiency of the proposed method has been proved by comparing CPU times of the RRC robot arm simulations from the proposed method and the recursive

Newton-Euler method. The solution accuracy of the proposed method has been also validated by comparing simulation results from the commercial multibody dynamics program DADS. Although the algorithm has been applied to an open chain system, it is expected that the algorithm can be extended for closed loop systems without having any difficulty, as shown in RRC arm grabbing simulation.

References

- (1) Schiehlen, W., *Multibody Systems Handbook*, (1990), Springer-Verlag.
- (2) Haug, E.J., *Computer Aided Kinematics and Dynamics of Mechanical System*, (1989), Allyn and Bacon.
- (3) Haug, E.J., *Intermediate Dynamics*, (1992), Prentice-Hall.
- (4) Eich-Scellner, E. and Fuhrer, C., *Numerical Methods in Multibody Dynamics*, (1998), Teubner, Stuttgart.
- (5) Wang, L.T. and Ravani, B., *Recursive Computations of Kinematic and Dynamic Equations For Mechanical Manipulators*, IEEE J. of Robotics and Automation, Vol. RA-1, No. 3 (1985), pp. 124-131.
- (6) Blajer, W. and Schiehlen, W., *Efficient Determination of Constraint Reactions in Multibody Systems*, Proceedings of EUROMECH 427, LMT-Cachan, France. Sept. 24-27, (2001).
- (7) Kim, S.S. and Haug, E.J., *A Recursive Formulation for Flexible Multibody Dynamics, Part I: Open Loop Systems*, Computer Methods in Applied Mechanics and Engineering, Vol. 71 (1988), pp. 293-314.
- (8) Tsai, F.F. and Haug, E.J., *Real-Time Multibody System Dynamic Simulation, Part II-A Parallel Algorithm and Numerical Results*, Mechanics of Structures and Machines, Vol. 19, No. 1 (1991), pp. 99-127.
- (9) Atkinson, K., *Numerical Analysis*, (1988), John Wiley & Sons.
- (10) Chuang, L.P. and Kim, S.S., *Real-Time Dynamics Simulation of a Robot System with Rotor Dynamics*, Proceedings of The ASME Design Technical Conferences, Advanced in Design Automation, DE-Vol. 23, No. 2 (1990), pp. 159-168.
- (11) DADS User's Manual, Revision 8.5, (1993), Computer Aided Design Software, Inc, Oakdale, Iowa.
- (12) Wittenburg, J., *Dynamics of Systems of Rigid Bodies*, (1977), B.G. Teubner Stuttgart.